UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/689,896	10/21/2003	James E. McCormick JR.	10982165-2	3875

22879 7590 06/12/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
06/12/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/689,896	MCCORMICK ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *10/21/2003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-12 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS as received on 10/21/2003.

Priority

3. Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, or 365(c) is acknowledged.

Specification

4. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

5. Applicant included two distinct claims labeled as claim 4. Consequently, the claims starting at the second claim 4 are misnumbered. Consequently, the examiner has renumbered claims 1-11 as claims 1-12, respectively. Applicant is asked to correct this misnumbering.

6. Claim 2 is objected to because of the following informalities: Please insert --to-- after "adjacent". Appropriate correction is required.

7. Claim 5, due to renumbering, is objected to because it should now be dependent on claim 4. Appropriate correction is required. It is unclear if applicant wishes to also change the dependency of claims 6 and 7 to be dependent on claim 4 or if they should remain dependent on claim 3.

8. Claim 9, due to renumbering, is objected to because it should now be dependent on claim 8. Appropriate correction is required.

9. Claim 12, due to renumbering, is objected to because it should now be dependent on claim 11. Appropriate correction is required.

Double Patenting

10. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

11. Claims 1-12 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 1, 1, 3, 4, 5, 8, 6, 7, 3, 4, and 4, respectively, of U.S. Patent No. 6,721,875 (hereafter '875).

12. Regarding claim 1 of the instant application, although the conflicting claims are not identical, they are not patentably distinct from each other because claim 1 of the instant application is anticipated by claim 1 of '875. That is, claim 1 of '875 includes all limitations of claim 1 of the instant application, and therefore, claim 1 of the instant application is unpatentable due to obvious-type double-patenting. It should be noted that the last paragraph of claim 1 of the instant application, while worded differently from claim 1 of '875, is still anticipated by claim 1 of '875 because if the first branch execution unit adds N and P offsets to the pointer in claim 1 of '875 (which is the case), then the P offset must be accessible by the first branch execution unit. Therefore, bits in the second syllable (which holds the P offset) are made available to the first branch execution unit, as claimed in claim 1 of the instant application.

13. Regarding claim 2 of the instant application, given that claim 1 of the instant application is anticipated by claim 1 of '875, claim 2 of the instant application is also anticipated by claim 1 of '875.

14. Regarding claim 3 of the instant application, given that claim 1 of the instant application is anticipated by claim 1 of '875, claim 3 of the instant application is also anticipated by claim 1 of '875.

15. Regarding claim 4 of the instant application, given that claim 3 of the instant application is anticipated by claim 1 of '875, claim 4 of the instant application is anticipated by claim 3 of '875.

16. Regarding claim 5 of the instant application, given that claim 3 of the instant application is anticipated by claim 1 of '875, claim 5 of the instant application is also anticipated by claim 4 of '875.

17. Regarding claim 6 of the instant application, given that claim 3 of the instant application is anticipated by claim 1 of '875, claim 6 of the instant application is also anticipated by claim 5 of '875.

18. Regarding claim 7 of the instant application, given that claim 3 of the instant application is anticipated by claim 1 of '875, claim 7 of the instant application is also anticipated by claim 8 of '875.

19. Regarding claim 8 of the instant application, given that claim 1 of the instant application is anticipated by claim 1 of '875, claim 8 of the instant application is also anticipated by claim 6 of '875.

20. Regarding claim 9 of the instant application, given that claim 8 of the instant application is anticipated by claim 6 of '875, claim 9 of the instant application is also anticipated by claim 7 of '875.

21. Regarding claim 10 of the instant application, although the conflicting claims are not identical, they are not patentably distinct from each other because claim 10 of the instant application is anticipated by claim 3 of '875. That is, claim 3 of '875 includes all limitations of claim 10 of the instant application, and therefore, claim 10 of the instant application is unpatentable due to obvious-type double-patenting. Note that the multiplexer details of claim 10 of the instant application are broader in scope than the multiplexer details of claim 3.

22. Regarding claim 11 of the instant application, although the conflicting claims are not identical, they are not patentably distinct from each other because the processor of claim 4 of'875 performs the method of claim 11 of the instant application. That is, the processor of claim 4 of '875 performs all steps set forth in claim 11 of the instant application. Since a processor performs a method, they are obvious variations of one another. Therefore, claim 11 of the instant application is unpatentable due to obvious-type double-patenting.

23. Regarding claim 12 of the instant application, given that claim 11 of the instant application is anticipated by claim 4 of '875, claim 12 of the instant application is also anticipated by claim 4 of '875.

Claim Rejections - 35 USC § 103

24. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

25. Claims 1-3, 6, 8-9, and 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yoshida, U.S. Patent No. 5,761,470, in view of Sharangpani et al., U.S. Patent No. 6,237,077 (herein referred to as Sharangpani).

26. Referring to claim 1, Yoshida has taught a processor, comprising:

a) an instruction set comprising a single-syllable IP-relative branch instruction and a long IP-relative branch instruction; the long IP-relative branch instruction occupying multiple syllables of an instruction bundle. First, see Fig.3, component 31. The SB (short branch) format is a short

IP-relative branch format that includes a displacement of 11 bits. Also, note that the SB instruction format is 20 bits in length, which would allow it to fit in a single one of the 20-bit instruction syllables shown in Fig.1. Next, see Fig.4, component 71. The LB (long branch) format is a long IP-relative branch format that includes a displacement of 29 bits. Note that the LB instruction format is 40 bits in length, which would require it to occupy two of the 20-bit syllables as shown in Fig.1. And, since the displacement is 29 bits in length and each syllable only holds 20 bits, one syllable will be used to hold a first amount of offset bits while a second syllable will be used to hold a second amount of offset bits.

b) a first branch execution unit, the first branch execution unit comprising an adder to A) in a first mode, calculate a branch target of a single-syllable IP-relative branch instruction located in a first syllable of an instruction bundle, and B) in a second mode, calculate a branch target of a long IP-relative branch instruction located in said first syllable and a second syllable of an instruction bundle. See column 10, lines 43-51, and note that for a branch instruction (both short and long), the displacement/offset is added, by the first branch execution unit, to the current contents of the program counter. Therefore, in the case of the single-syllable (i.e., short) branch instruction, an M-bit offset would be added to the PC. And, in the case of the multiple-syllable (i.e., long) branch instruction, an X-bit offset would be added to the PC, where the X-bit offset comprises N bits from a first syllable and P bits from a second syllable.

c) Yoshida has not taught a second branch execution unit to calculate a branch target of a single-syllable IP-relative branch instruction located in said second syllable of an instruction bundle. However, Sharangpani has taught the concept of at least a second branch execution (BR2) adjacent to the first branch execution unit (BR1). See Fig.3. As disclosed by Sharangpani in

column 7, lines 25-27, by employing multiple branch execution units, multiple branches can be executed in parallel. A person of ordinary skill in the art would have recognized that if each branch generally takes 1 clock cycle to execute, then a parallel execution of three branches would still take just 1 clock cycle. On the other hand, with serial execution, execution of three branch instructions would require at least 3 clock cycles. As a result, parallel execution reduces the amount of time required to execute branch instructions, especially when a second branch within a single bundle is the only branch that is taken. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida's system to include a second branch execution unit, as taught by Sharangpani, in order to achieve better execution efficiency. Sharangpani, modified as above, now has the ability to calculate a first branch target of a first branch instruction in a first syllable, and calculate a second branch target of a second branch instruction in a second syllable, as claimed.

d) Yoshida in view of Sharangpani has not explicitly taught wires, routed over the first branch execution unit, to provide bits in said second syllable of an instruction bundle to both the first and second branch execution units. However, recall that it has already been established that wires connect the second syllable to the second branch unit because the second branch unit is to calculate a second branch target of a second branch instruction in a second syllable. The reason for the second branch execution unit is to allow calculation of multiple branch targets in parallel. However, when a single long branch exists, only a single branch target needs to be calculated (from a combination of offsets in multiple syllables), and therefore, only a single adder (branch execution unit) is needed. One of ordinary skill in the art would've recognized that:

a) using both adders in atypical fashion to calculate a single target address, when just one

adder is perfectly capable of accomplishing such a task, would be a waste of power, as more circuitry must be powered than required (two adders as opposed to one); and

b) either the first or second adder (branch execution unit) could be used to calculate the target as both of them perform the same function. Consequently, one of ordinary skill in the art would have also recognized that the first branch execution unit would be able to perform the addition, and would this require wires routed over the first execution unit so that the partial offset from a second syllable could be provided to the first execution unit.

As a result, in order to save power while calculating a target address of a long branch instruction, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida in view of Sharangpani to include wires, routed over the first branch execution unit, to provide bits in said second syllable of an instruction bundle to both the first and second branch execution units.

27. Referring to claim 2, Yoshida in view of Sharangpani has taught the processor of claim 1, wherein the first and second branch execution units are physically adjacent each other. See Sharangpani, Fig.3.

28. Referring to claim 3, Yoshida in view of Sharangpani has taught the processor of claim 1, wherein:

a) single-syllable IP-relative branch instructions carry M offset bits. See Yoshida, Fig.1, and note that a single syllable (for instance, component 12) may hold a single-syllable branch instruction 31 shown in Fig.3, where the single-syllable branch instruction includes an offset of M=11 bits.

b) long IP-relative branch instructions carry N offset bits in said first syllable of an instruction bundle, and P offset bits in said second syllable of an instruction bundle. See Yoshida, Fig.1, and note that multiple syllables 15 may hold a long branch instruction 71 shown in Fig.4, where the 29-bit offset is clearly divided between syllables into N and P-bit portions.

c) the wires alternately carry said M offset bits of single-syllable IP-relative branch instructions carried in said second syllable of an instruction bundle, or said P offset bits of a long IP-relative branch instruction. As described above, if the first branch execution unit calculates a branch target for the branch in syllable 12 of Fig.1, then that execution unit will receive the M bit offset when a short branch exists in that syllable. However, when a long branch exists in syllables 15, then the P bits from the second syllable (rightmost 20 bits of syllables 15) will be routed on the wires to the first execution unit.

29. Referring to claim 6, Yoshida in view of Sharangpani has taught the processor of claim3 wherein a merger of the N and P offset bits of a long IP-relative branch instruction provide an offset which, when added to an instruction pointer value, is capable of redirecting an instruction pointer stored in the processor's address space. From Yoshida, column 7, lines 1-9, it can be seen that all instruction addresses end in "000". It is further disclosed that the 3 bits are added to the branch reach. For instance the 11-bit offset of the short branch becomes a 14-bit branch offset due to the fact that the three low bits of the instruction addresses are always "000". Likewise the 29-bit offset of the long branch actually becomes a 32-bit branch offset with the addition of the 3 extra bits. Therefore, since the long branch offset is equivalent to 32-bits and the program counter can hold a value that can select any address in a 32-bit value address space,

then it follows that the long branch offset (N+P bits) allows for the redirection of the program to any instruction in the address space.

30. Referring to claim 8, Yoshida in view of Sharangpani has taught the processor of claim 1, wherein each instruction bundle consists of three instruction syllables. See Yoshida, Fig.1.

31. Referring to claim 9, Yoshida in view of Sharangpani has taught the processor of claim 8. Yoshida has further taught that each instruction bundle further comprises a template field, wherein one state of the template field maps a first syllable of an instruction bundle to a memory execution unit, and maps second and third syllables of an instruction bundle to a branch execution unit, the branch execution unit being configured to calculate a target of along IP-relative branch instruction. See Fig.1, and note the "FM" field, which specifies the format of the instruction bundle. The "FM" field is set to a specific value, depending on the contents of the corresponding instruction bundle. See Fig.2. For instance, when FM = 0000, the short instruction in field 14 of Fig.1 will be issued to its corresponding functional unit at the same time the long instruction of field 15 is issued to its functional unit. From Fig.3, it can be seen that there are short memory instructions, SM-1 and SM-2 (see column 7, lines 24-25). Also, there are long branch instructions of format LB in Fig.4. Therefore, the first syllable can hold a memory instruction which would be mapped to the memory unit 125 (Fig.10) while the second and third syllables would hold a long branch instruction, which will be mapped to the PC unit 121 (Fig.10) in order to calculate a branch target.

32. Referring to claim 11, Yoshida has taught a method, comprising:

a) routing to a first branch execution unit, wires that carry bits of an instruction syllable of an instruction bundle. See Fig.3, instruction 31, and Fig.4, instruction 71. These short and long

branch instructions, respectively, include offsets/displacements that are added to the program counter to realize a branch target address. Consequently, wires must carry the offset from the instruction syllable(s) (Fig.1) to the execution unit which adds the program counter and the offset. See column 10, lines 43-51.

b) Yoshida has not taught a second branch execution unit (and wires routed to it). However, Sharangpani has taught the concept of at least a second branch execution (BR2) adjacent to the first branch execution unit (BR1). See Fig.3. As disclosed by Sharangpani in column 7, lines 25-27, by employing multiple branch execution units, multiple branches can be executed in parallel. A person of ordinary skill in the art would have recognized that if each branch generally takes 1 clock cycle to execute, then a parallel execution of three branches would still take just 1 clock cycle. On the other hand, with serial execution, execution of three branch instructions would require at least 3 clock cycles. As a result, parallel execution reduces the amount of time required to execute branch instructions, especially when a second branch within a single bundle is the only branch that is taken. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida's system to include a second branch execution unit (and wires routed to it), as taught by Sharangpani, in order to achieve better execution efficiency. Sharangpani, modified as above, now has the ability to calculate a first branch target of a first branch instruction in a first syllable, and calculate a second branch target of a second branch instruction in a second syllable.

c) Yoshida in view of Sharangpani has not taught that when the first branch execution unit is configured to calculate a branch target of a long IP-relative branch instruction occupying multiple syllables of an instruction bundle, coupling the wires to the first branch execution unit,

and otherwise not coupling the wires to the first branch execution unit. However, recall that it has already been established that wires connect the second syllable to the second branch unit because the second branch unit is to calculate a second branch target of a second branch instruction in a second syllable. The reason for the second branch execution unit is to allow calculation of multiple branch targets in parallel. However, when a single long branch exists, only a single branch target needs to be calculated (from a combination of offset portions in multiple syllables), and therefore, only a single adder (branch execution unit) is needed. One of ordinary skill in the art would've recognized that:

1) using both adders in atypical fashion to calculate a single target address, when just one adder is perfectly capable of accomplishing such a task, would be a waste of power, as more circuitry must be powered than required (two adders as opposed to one); and

2) either the first or second adder (branch execution unit) could be used to calculate the target as both of them perform the same function. Consequently, one of ordinary skill in the art would have also recognized that the first branch execution unit would be able to perform the addition, and would this require wires routed over the first execution unit so that the partial offset from a second syllable could be provided to the first execution unit.

As a result, in order to save power while calculating a target address of a long branch instruction, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida in view of Sharangpani such that the wires are routed over the first branch execution unit which is configured to calculate a branch target of a long IP-relative branch instruction occupying multiple syllables of an instruction bundle, and when it is to perform such a calculation, the wires from the second syllable are coupled to the first branch

execution unit (i.e., the first branch execution unit will receive an offset from the second syllable), and otherwise not coupling the wires to the first branch execution unit (i.e., the first branch execution unit will not receive an offset from the second syllable).

33. Referring to claim 12, Yoshida in view of Sharangpani has taught a method as described in claim 11.

a) Yoshida has further taught that when a branch target of a long IP-relative branch instruction is being calculated by the first branch execution unit, filling a first number of bit positions of an addend input of an adder of said first branch execution unit with bits carried over said wires, and filling a second number of bit positions of the addend input of said adder with additional offset bits supplied by a second instruction syllable of a long IP-relative branch instruction. See Fig.1 and Fig.4, component 71. Note that the long branch displacement if too large to fit into a single syllable. Consequently, it spills into a second syllable as shown in component 15 of Fig.1. When the branch target for the long branch is to be calculated, both offset portions (from the multiple syllables) must be provided to the adder.

b) Yoshida has further taught that when a branch target of a single-syllable IP-relative branch instruction is being calculated by the first branch execution unit, filling the first number of bit positions of the addend input of said adder with offset bits supplied by a single-syllable IP-relative branch instruction, and filling the second number of bit positions of the addend input of said adder with a number of sign extension bits. Note from Fig.5 that sign extension is performed, and hence, the system utilizes signed numbers. It should be realized that the program counter is a 32-bit number as seen from the 32-bit PC shown in Fig.7. It should be further realized that in order to correctly add two numbers together in a system with sign bits, the

smaller number should be sign-extended to the size of the larger number. For instance, suppose the PC were a 5-bit value, and its current value was 00001 (equal to 1 in signed notation), and a three bit signed displacement with a value of 101 (equal to -3 in signed notation) had to be added to it, the following would happen: If the displacement 101 was not sign extended to 11101, the addition would be 00001 + 00101 = 00110 = 6, which is incorrect since -3 + 1 = -2. On the other hand, with sign extension, the addition would be 00001 + 11101 = 11110 = -2, which is incorrect. Consequently, since the M-bit displacement of the short branch is only 11 bits wide, it must be sign-extended before it is added to the PC.

34. Claims 4-5 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yoshida in view of Sharangpani, and further in view of Wang et al., U.S. Patent No. 4,935,867 (herein referred to as Wang).

35. Referring to claim 4, Yoshida in view of Sharangpani has taught a processor as described in claim 3. Yoshida in view of Sharangpani has not taught a multiplexer comprising:

a) first data inputs coupled to receive a number of sign extension bits and second data inputs coupled to receive the P offset bits of a long IP-relative branch instruction; wherein, when the first branch execution unit is configured in the first mode, the multiplexer outputs the number of sign extension bits which are received at its first data inputs, and said number of sign extension bits are merged with the M offset bits of a single-syllable IP-relative branch instruction to form at least part of said addend which is added to an instruction pointer value; and wherein, when the first branch execution unit is configured in the second mode, the multiplexer outputs the P offset bits of a long IP-relative branch instruction which are received at its second data inputs, and said

P offset bits are merged with the N offset bits of the same long IP-relative branch instruction to form at least part of said addend which is added to an instruction pointer value. However, Wang has taught the idea of using a multiplexer to determine which of two inputs will be propagated to an adder. See Fig.1A, component 66. The multiplexer of Wang comprises:

1) first data inputs coupled to receive a number of sign extension bits. See component 64 and note that three offset bits (i.e. a short offset) are sign extended and supplied to the multiplexer.

2) second data inputs coupled to receive a larger offset. See Fig.1A, component 62. It should be realized by the applicant that the basic concept of supplying offset bits via multiplexer has been taught. Therefore, although Wang has not explicitly taught that the P offset bits of a long-branch instruction are coupled to second data inputs of a multiplexer, a person of ordinary skill in the art at the time of the invention would have recognized that the multiplexer could be used to propagate P offset bits in the same fashion as Wang. It should be further realized that the index register (component 62 of Fig.1A) could be thought of as storage for the syllable that holds the P offset bits.

Since two different branch instruction formats exist (long and short) within Yoshida's system, two different approaches must be taken when encountering each of these branches. When the short branch is encountered, a person of ordinary skill in the art would've recognized that before adding a short offset of M bits to the PC, the M bits should be sign-extended (Fig.5 of Yoshida) to create a correct value equal in length to the PC so that addition occurs properly. On the other hand, when the long branch is encountered, the N and P bits should be merged to make up at least part of the addend to be added to the instruction pointer. Therefore, since two

different modes are required based on the instruction type, a multiplexer can be used to select between two different data sources, one being the sign extension bits needed for the short branch, and the other being the P offset bits which are needed for the long branch. Wang has taught the functionality of providing both an offset and sign bits through a multiplexer. Therefore, it would have been obvious to one of ordinary skill in the art to implement the multiplexer as taught by Wang into Yoshida's system in order to provide an efficient way of propagating two different data sources depending on the type of instruction encountered.

36. Referring to claim 5, Yoshida in view of Sharangpani and further in view of Wang has taught the processor of claim 3, wherein the wires are coupled to the second data inputs of the multiplexer. Since the second input of the mux receives the P offset, and the P offset comes from the second syllable via the wires routed over the first execution unit, then the mux receives as its second input, the wires.

37. Referring to claim 10, Yoshida has taught a processor, comprising:

a) an instruction set comprising a single-syllable IP-relative branch instruction and a long IP-relative branch instruction; the long IP-relative branch instruction occupying multiple syllables of an instruction bundle. First, see Fig.3, component 31. The SB (short branch) format is a short IP-relative branch format that includes a displacement of 11 bits. Also, note that the SB instruction format is 20 bits in length, which would allow it to fit in a single one of the 20-bit instruction syllables shown in Fig.1. Next, see Fig.4, component 71. The LB (long branch) format is a long IP-relative branch format that includes a displacement of 29 bits. Note that the LB instruction format is 40 bits in length, which would require it to occupy two of the 20-bit syllables as shown in Fig.1. And, since the displacement is 29 bits in length and each syllable

only holds 20 bits, one syllable will be used to hold a first amount of offset bits while a second syllable will be used to hold a second amount of offset bits.

b) a first branch execution unit. See column 10, lines 43-51, and note that for a branch instruction (both short and long), the displacement/offset is added, by the first branch execution unit, to the current contents of the program counter. Therefore, in the case of the single-syllable (i.e., short) branch instruction, an M-bit offset would be added to the PC. And, in the case of the multiple-syllable (i.e., long) branch instruction, an X-bit offset would be added to the PC, where the X-bit offset comprises N bits from a first syllable and P bits from a second syllable.

c) Yoshida has not taught a second branch execution unit. However, Sharangpani has taught the concept of at least a second branch execution (BR2) adjacent to the first branch execution unit (BR1). See Fig.3. As disclosed by Sharangpani in column 7, lines 25-27, by employing multiple branch execution units, multiple branches can be executed in parallel. A person of ordinary skill in the art would have recognized that if each branch generally takes 1 clock cycle to execute, then a parallel execution of three branches would still take just 1 clock cycle. On the other hand, with serial execution, execution of three branch instructions would require at least 3 clock cycles. As a result, parallel execution reduces the amount of time required to execute branch instructions, especially when a second branch within a single bundle is the only branch that is taken. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida's system to include a second branch execution unit, as taught by Sharangpani, in order to achieve better execution efficiency. Sharangpani, modified as above, now has the ability to calculate a first branch target of a first branch instruction in a first syllable,

and calculate a second branch target of a second branch instruction in a second syllable, as claimed.

d) Yoshida in view of Sharangpani has not explicitly taught wires, routed over the first branch execution unit, to provide bits in said second syllable of an instruction bundle to both the first and second branch execution units. However, recall that it has already been established that wires connect the second syllable to the second branch unit because the second branch unit is to calculate a second branch target of a second branch instruction in a second syllable. The reason for the second branch execution unit is to allow calculation of multiple branch targets in parallel. However, when a single long branch exists, only a single branch target needs to be calculated (from a combination of offsets in multiple syllables), and therefore, only a single adder (branch execution unit) is needed. One of ordinary skill in the art would've recognized that:

1) using both adders in atypical fashion to calculate a single target address, when just one adder is perfectly capable of accomplishing such a task, would be a waste of power, as more circuitry must be powered than required (two adders as opposed to one); and

2) either the first or second adder (branch execution unit) could be used to calculate the target as both of them perform the same function. Consequently, one of ordinary skill in the art would have also recognized that the first branch execution unit would be able to perform the addition, and would this require wires routed over the first execution unit so that the partial offset from a second syllable could be provided to the first execution unit.

As a result, in order to save power while calculating a target address of a long branch instruction, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida in view of Sharangpani to include wires, routed over the first

branch execution unit, to provide bits in said second syllable of an instruction bundle to both the first and second branch execution units.

e) Yoshida in view of Sharangpani has not taught a multiplexer comprising first data inputs coupled to receive a number of sign extension bits, and second data inputs coupled to receive from said wires offset bits of a long IP-relative branch instruction; the multiplexer outputting the number of sign extension bits to the first branch execution unit when the first branch execution unit is configured to calculate a branch target of a single IP-relative branch instruction, and the multiplexer outputting the offset bits of a long IP-relative branch instruction to the first branch execution unit when the first branch execution unit is configured to calculate a target for a long IP-relative branch instruction. However, Wang has taught the idea of using a multiplexer to determine which of two inputs will be propagated to an adder. See Fig.1A, component 66. The multiplexer of Wang comprises:

1) first data inputs coupled to receive a number of sign extension bits. See component 64 and note that three offset bits (i.e. a short offset) are sign extended and supplied to the multiplexer.

2) second data inputs coupled to receive a larger offset. See Fig.1A, component 62. It should be realized by the applicant that the basic concept of supplying offset bits via multiplexer has been taught. Therefore, although Wang has not explicitly taught that the P offset bits of a long-branch instruction are coupled to second data inputs of a multiplexer, a person of ordinary skill in the art at the time of the invention would have recognized that the multiplexer could be used to propagate P offset bits in the same fashion as Wang. It should be further realized that the

index register (component 62 of Fig.1A) could be thought of as storage for the syllable that holds the P offset bits.

Since two different branch instruction formats exist (long and short) within Yoshida's system, two different approaches must be taken when encountering each of these branches. When the short branch is encountered, a person of ordinary skill in the art would've recognized that before adding a short offset of M bits to the PC, the M bits should be sign-extended (Fig.5 of Yoshida) to create a correct value equal in length to the PC so that addition occurs properly. On the other hand, when the long branch is encountered, the N and P bits should be merged to make up at least part of the addend to be added to the instruction pointer. Therefore, since two different modes are required based on the instruction type, a multiplexer can be used to select between two different data sources, one being the sign extension bits needed for the short branch, and the other being the P offset bits which are needed for the long branch. Wang has taught the functionality of providing both an offset and sign bits through a multiplexer. Therefore, it would have been obvious to one of ordinary skill in the art to implement the multiplexer as taught by Wang into Yoshida's system in order to provide an efficient way of propagating two different data sources depending on the type of instruction encountered.

38. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Yoshida in view of Sharangpani.

39. Referring to claim 7, Yoshida in view of Sharangpani has taught the processor of claim 3. Yoshida has not taught that M=N. However, it should be noted that shifting locations of parts (e.g. bits), absent evidence of unexpected results due to the shifting, is generally not given

patentable weight or would have been obvious improvements. See In re Japikse, 86 USPQ 70 (CCPA 1950). Recall that the single-syllable branch is 20 bits and the long branch is 40 bits. For the single-syllable branch, all 11 (M) bits of offset are stored in a first syllable. A person of ordinary skill in the art would have recognized that the long branch instruction's bits could be rearranged in any way without affecting operation (as long as the system knows which bits are which), such that 11 bits of the long offset are stored in the first syllable and the remaining (P) offset bits are stored in a second syllable. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshida's system such that M=N.

Conclusion

40. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Ohshima et al., U.S. Patent No. 5,634,136, has taught a VLIW system in which a long immediate filed is divided among multiple syllables and concatenated when needed during processing.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJH
David J. Huisman
May 24, 2007